

Contoh Modul Ajar Fase F

Informasi Umum

Mata Pelajaran : Informatika
Fase : F
Kelas : XI
Alokasi Waktu : 6 JP x 40 menit (2 pertemuan)

Modul ini menggambarkan kegiatan pembelajaran yang memfasilitasi murid untuk mampu menuliskan algoritma yang efisien, efektif dan optimal. Kegiatan pembelajaran diharapkan mampu memberikan gambaran dan pemahaman kepada murid untuk mampu memahami dan menuliskan algoritma yang efisien, efektif dan optimal yang berkorelasi dengan kehidupan sehari-hari mereka.

Dimensi Profil Lulusan

- Penalaran Kritis
- Kreativitas
- Kolaborasi
- Kemandirian
- Komunikasi

Tujuan Pembelajaran

Menuliskan algoritma yang efisien, efektif, dan optimal.

Praktik Pedagogis

pembelajaran berbasis proyek

Lingkungan Pembelajaran

- Ruang Fisik: Pembelajaran dilaksanakan di ruang kelas yang memungkinkan pembagian kelompok secara fleksibel. Kegiatan ini juga melibatkan kerja kolaboratif seperti berdiskusi, menuliskan algoritma, serta mempresentasikan hasil kerja kelompok
- Budaya Belajar: Pembelajaran membangun budaya kolaboratif dan apresiatif

Kemitraan Pembelajaran

Pemanfaatan Digital

Pemanfaatan digital dilakukan secara fungsional melalui penggunaan Canva untuk membuat dan menyampaikan hasil kerja murid

Langkah Kegiatan Pembelajaran

Pertemuan 1

Mengenal konsep dasar Algoritma (3 JP x 45 menit)



Memahami (berkesadaran, menggemirakan)

1. Pendidik meminta murid untuk membuat urutan alur atau langkah-langkah menjadwalkan antrian murid dalam proses mengantri makanan di kantin agar terjadi antrian yang tertib untuk mengetahui kesiapan belajar, kebutuhan, dan level pencapaian murid terkait dengan menuliskan mengemukakan algoritma yang efektif, efisien dan optimal serta ditanggapi secara langsung.

Profil Murid

Pemula	Menengah	Mahir
Murid yang belum mampu menuliskan alur urutan.	Murid yang mampu menuliskan alur urutan yang sistematis.	Murid yang pada tiap poin mampu menuliskan alur urutan yang sistematis, efisien (tidak banyak pengulangan), efektif, optimal, dan benar

2. Pendidik memberikan pertanyaan pemantik berupa:
 - a. Bagaimana jika alur yang anda sebutkan tidak berurut?
 - b. Apakah alur yang anda buat tidak ada pengulangan yang tidak dibutuhkan?
 - c. Hal-hal apa saja yang perlu ada dalam sebuah penulisan algoritma? (Pendidik memberikan stimulus yang mengarah ke ciri-ciri algoritma).
3. Melakukan pembagian kelompok dengan membagi proporsi kelompok, dimana di setiap kelompok diharapkan ada murid dari kategori pemula, menengah, dan mahir supaya murid dengan kategori mahir bisa menjadi tutor sebaya untuk rekan di kelompoknya.
4. Pendidik memfasilitasi proses tukar gagasan antar kelompok dengan memberikan pertanyaan tentang pengertian algoritma dan menuliskan alur atau langkah-langkah dari contoh-contoh implementasi algoritma dalam kehidupan sehari-hari murid melalui canva. https://bit.ly/algoritma_fomatif1



5. Setiap murid saling memberikan komentar terhadap pendapat murid dari kelompok lain.
6. Pendidik mengobservasi aktivitas yang dilakukan murid sebagai bagian dari asesmen formatif.



Merefleksi (berkesadaran, bermakna, menggembirakan)

1. Pendidik memberikan apresiasi terhadap hasil kerja kelompok beserta pendapat murid dan memberikan umpan balik diberikan secara langsung.
2. Pendidik memberikan penguatan dan menjelaskan tentang pengertian algoritma menjelaskan langkah yang dapat membantu murid menuliskan algoritma yang efisien, efektif dan optimal menggunakan diagram alir (*flowchart*), notasi algoritma deskriptif dan *pseudocode*. Beberapa metode yang dapat diberikan kepada murid untuk membantu mereka mencapai pemahaman ini dilakukan melalui pendekatan bertahap ([lihat bahan ajar penulisan algoritma](#)):
 - a. Mulai dengan contoh sederhana: Mulailah dengan contoh-contoh sederhana yang mudah dipahami oleh murid. Hal ini dapat membantu mereka untuk memahami konsep dasar algoritma dan bagaimana menulisnya dengan cara yang terstruktur.
 - b. Meningkatkan kompleksitas secara bertahap: Setelah murid memahami konsep dasar, berikan mereka contoh-contoh yang lebih kompleks dengan langkah-langkah yang lebih banyak dan pernah mereka alami dalam kehidupan sehari-hari. Hal ini akan membantu mereka untuk mengembangkan kemampuan mereka dalam memecahkan masalah yang lebih besar.

- c. Gunakan visualisasi: Gunakan visualisasi seperti diagram alir (*flowchart*) untuk membantu murid memahami struktur algoritma. Hal ini dapat membantu mereka untuk melihat bagaimana langkah-langkah dalam algoritma saling terkait.
 - d. Contoh visualisasi menggunakan diagram alir (*flowchart*).
3. Memberikan umpan balik dan menjelaskan manfaat memahami konsep algoritma dalam kehidupan sehari-hari.

Rubrik Penilaian Mengenal konsep dasar Algoritma

Asesmen Formatif

Kriteria	Cukup	Baik	Sangat Baik
Pemahaman Pengertian Algoritma	Tidak mampu menjelaskan pengertian algoritma.	Mampu menjelaskan pengertian algoritma	Mampu menjelaskan pengertian algoritma beserta pendapat dari beberapa ahli
Pemahaman Konsep Algoritma	Mampu menuliskan alur urutan yang sistematis namun belum efisien	Mampu menuliskan alur urutan yang sistematis dan efisien	Mampu menuliskan alur urutan yang sistematis dan efisien serta menunjukkan ciri-ciri yang terdapat pada alur.
Diskusi Kelompok	Kadang terlibat dalam diskusi	Aktif terlibat dalam diskusi	Solutif dan akomodatif dalam diskusi

Rencana Tindak Lanjut

1. Pendidik perlu memperhatikan dengan benar setiap indikator yang dinilai.
2. Kondisi berwarna biru, menunjukkan kondisi yang diharapkan pada pembelajaran kali ini. Apabila terdapat murid yang tidak menunjukkan kondisi tersebut maka pendidik dapat memberikan intervensi sebagai berikut:

Kriteria	Bentuk Intervensi
Pemahaman Pengertian Algoritma	Memberikan penguatan materi dengan memberikan beberapa sumber belajar.
Pemahaman Konsep Algoritma	Memberikan tugas tambahan sebagai penguatan pemahaman konsep.
Diskusi Kelompok	Memberikan pembinaan.

3. Apabila murid sudah mencapai kondisi yang diharapkan pendidik dapat memberikan apresiasi atau tantangan pada pembelajaran selanjutnya.

Pertemuan 2

Penulisan algoritma dengan Notasi Algoritma Deskriptif, *Pseudocode* dan Diagram Alir (*Flowchart*) (3 JP x 45 menit)



Mengaplikasi (bermakna, menggembirakan)

1. Murid secara berkelompok menuliskan algoritma dari aktivitas yang telah ditentukan pada lembar kerja murid (LKPD 1) dan merancang presentasinya. https://bit.ly/algoritma_lkpd
2. Pendidik memberikan pendampingan konstruktif sepanjang penulisan algoritma dan presentasi dengan memberikan tips-tips ringan pada murid.
3. Murid secara bergantian mempresentasikan hasil diskusi kelompoknya. Pendidik menilai presentasi dan hasil penulisan algoritmanya sebagai bagian dari asesmen sumatif.
4. Murid menarik kesimpulan bersama dengan dipandu oleh pendidik.

Refleksi Murid

- a. Merefleksi perasaan setelah pembelajaran.
- b. Merefleksi materi yang telah dikuasai.
- c. Merefleksi materi yang belum dikuasai.

Rubrik Penilaian Penulisan Algoritma dengan Diagram Alir (*Flowchart*), Notasi Algoritma Deskriptif dan *Pseudocode*

Asesmen Sumatif

Kriteria	Perlu Perbaikan	Cukup	Baik	Sangat Baik
Diagram Alir (<i>Flowchart</i>) 1. Alur yang dibuat sistematis 2. Simbol yang digunakan sesuai 3. Variabel yang digunakan logis 4. Alur yang dibuat efektif	Menunjukkan 1 komponen	Menunjukkan 2 komponen	Menunjukkan 3 komponen	Menunjukkan 4 komponen

Kriteria	Perlu Perbaikan	Cukup	Baik	Sangat Baik
Notasi Algoritma Deskriptif 1. Penulisan yang dibuat sistematis 2. Penulisan jelas dan mudah dipahami. 3. Penulisan alur dibuat efektif 4. Terdapat kesesuaian dengan diagram alir.	Menunjukkan 1 komponen	Menunjukkan 2 komponen	Menunjukkan 3 komponen	Menunjukkan 4 komponen
<i>Pseudocode</i> 1. Penulisan yang dibuat sistematis 2. Penulisan sesuai dengan kondisi. 3. Penulisan alur dibuat efektif 4. Terdapat kesesuaian dengan diagram alir.	Menunjukkan 1 komponen	Menunjukkan 2 komponen	Menunjukkan 3 komponen	Menunjukkan 4 komponen
Aktivitas Kelompok 1. Menampilkan sajian yang menarik 2. Berkomitmen pada tujuan yang ingin dicapai sesuai tugas dan peran 3. Penyampaian yang lugas saat presentasi 4. Aktif terlibat dalam diskusi	Menunjukkan 1 komponen	Menunjukkan 2 komponen	Menunjukkan 3 komponen	Menunjukkan 4 komponen

Rencana Tindak Lanjut

1. Pendidik perlu memperhatikan dengan benar setiap indikator yang dinilai.
2. Kondisi berwarna biru, menunjukkan kondisi yang diharapkan pada pembelajaran kali ini. Apabila terdapat murid yang tidak menunjukkan kondisi tersebut maka pendidik dapat memberikan intervensi sebagai berikut:

Kriteria	Bentuk Intervensi
Diagram Alir (<i>Flowchart</i>)	Memberikan pendampingan di luar jam pelajaran dengan terlebih dahulu membuat kesepakatan dengan murid tersebut dengan menggunakan lembar kerja murid (LKPD 2) yang telah disesuaikan. https://bit.ly/algoritma_lkpd
Notasi Algoritma Deskriptif	
<i>Pseudocode</i>	
Aktivitas kelompok	Memberikan pembinaan.

3. Apabila murid sudah mencapai kondisi yang diharapkan pendidik dapat memberikan apresiasi atau tantangan pada pembelajaran selanjutnya.

Bahan Ajar

Algoritma

Algoritma adalah suatu kumpulan instruksi terstruktur dan terbatas yang dapat diimplementasikan dalam bentuk program komputer untuk menyelesaikan suatu permasalahan komputasi tertentu. Algoritma merupakan bentuk dari suatu strategi atau 'resep' yang kalian gunakan untuk menyelesaikan suatu masalah. Algoritma lahir dari suatu proses berpikir komputasional oleh seseorang untuk menemukan solusi dari suatu permasalahan yang diberikan. Dengan demikian, berpikir komputasional merupakan keahlian yang kalian perlukan untuk dapat membuat algoritma, program, atau suatu karya informatika yang dapat digunakan dengan efektif dan efisien.

A. Pengertian Algoritma Menurut Para Ahli

Beberapa ahli juga mengungkapkan atau memberikan pengertian dari algoritma. Di bawah ini akan dijelaskan beberapa pengertian algoritma dari beberapa ahli.

1 Abu Ja'far Muhammad Ibnu Musa Al-Khawarizmi

Abu Ja'far Muhammad Ibn Musa Al Khwarizmi mengatakan bahwa algoritma adalah sebuah cara atau metode khusus yang dapat digunakan untuk menyelesaikan satu atau beberapa masalah.

2 Marvin Minsky

Marvin Minsky merupakan seorang ahli *Artificial Intelligence* (AI) atau kecerdasan buatan berpendapat bahwa algoritma adalah suatu perangkat yang berbentuk aturan yang dapat menginformasikan kepada kita dari satu waktu ke waktu lainnya dan informasi yang diberikan merupakan bagaimana cara untuk bertindak.

3 Sismoro

Menurut Sismoro, algoritma adalah sekumpulan instruksi atau langkah-langkah yang sudah dituliskan secara sistematis dan digunakan untuk menyelesaikan suatu persoalan atau suatu permasalahan matematika dan logika dengan bantuan komputer.

4 Seymour Lipschutz dan Marc Lipson

Seymour Lipschutz dan Marc Lipson menyatakan bahwa algoritma adalah sebuah daftar yang berisi langkah demi langkah yang terhingga yang berasal dari berbagai macam perintah

yang sudah dijelaskan supaya bisa digunakan untuk menyelesaikan atau memecahkan suatu permasalahan yang ada.

5 Kani

Menurut Kani, algoritma adalah suatu usaha dengan sebuah urutan operasi yang sudah disusun secara sistematis dan logis dan dapat dimanfaatkan untuk menyelesaikan sebuah permasalahan demi menciptakan sebuah *output* tertentu.

6 S. E. Goodman dan S.T. Hedetniemi

Menurut S. E. Goodman dan S.T. Hedetniemi, algoritma adalah suatu urutan atau susunan yang sifatnya terbatas dari berbagai operasi yang sudah bisa dijelaskan serta setiap operasi membutuhkan waktu dan memori yang terbatas untuk memecahkan suatu permasalahan tertentu.

7 Donald Ervin Knuth

Donald Ervin Knuth mengatakan bahwa algoritma adalah kumpulan aturan-aturan yang berhingga dan bisa memberikan serangkaian operasi agar dapat memecahkan suatu permasalahan yang sedang terjadi.

B. Ciri-Ciri Algoritma

Segala macam metode yang ada pasti memiliki ciri-ciri termasuk algoritma. Berdasarkan apa yang diungkapkan Donald E. Knuth, algoritma memiliki beberapa ciri, yaitu:

1 Ada Input

Harus ada *Input* bisa diartikan sebagai setiap masalah yang dihadapi kedepannya harus dicarikan solusi agar masalah dapat diselesaikan dengan baik. Di dalam algoritma, minimal terdiri dari nilai 0 atau memiliki nilai lebih.

2 Ada Output

Harus ada *output* bisa dikatakan sebagai sebuah solusi dari suatu permasalahan yang sedang dihadapi. Di dalam algoritma, minimal harus ada 1 *output* atau lebih.

3 Adanya Sebuah Proses

Algoritma harus memiliki sebuah proses atau sekumpulan langkah-langkah yang harus dilakukan agar bisa menyelesaikan masalah atau mencapai tujuan akhir.

4 Instruksi yang Jelas

Algoritma akan berjalan dengan baik selama diberikan instruksi dan notasi yang disepakati, sehingga suatu kesalahan dapat diminimalisir dan berhasil menciptakan *output* yang baik.

5 Memiliki Tujuan Akhir

Sudah pasti kalau algoritma harus memiliki tujuan akhir. Dengan adanya tujuan akhir, kita akan berhenti setelah mencapai tujuan akhir.

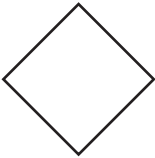
C. Penulisan Algoritma

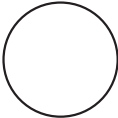
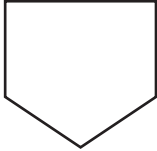
Penulisan algoritma dapat direpresentasikan dalam 3 notasi yaitu:

1 Diagram Alir (*Flowchart*)

Diagram alir dibuat dalam bentuk aliran simbol yang dapat ditelusuri dari suatu titik permulaan hingga titik akhir dari program. Diagram alir dibuat menggunakan simbol standar ANSI/ISO yang beberapa simbol dasarnya diberikan pada Tabel 7.1.

Tabel 7.1 Simbol Diagram Alir Beserta Maknanya

Simbol	Nama	Deskripsi
	Garis alir (<i>flowline</i>)	Arah yang menunjukkan aliran program dari awal hingga akhir
	Terminator	Titik awal atau titik akhir suatu program.
	Proses	Suatu kegiatan komputasi yang dilakukan oleh program: misalnya operasi aritmatika.
	Keputusan	Merupakan titik percabangan yang salah satu cabangnya dapat dilalui oleh program berdasarkan suatu kondisi.

Simbol	Nama	Deskripsi
	Masukan (<i>Input</i>)/ Keluaran (<i>Output</i>)	Melambangkan titik saat program akan menerima suatu data atau menghasilkan suatu informasi.
	Subprogram	Melambangkan suatu kegiatan atau proses lain yang telah didefinisikan sebelumnya.
	Penghubung dalam Halaman	Digunakan untuk menghubungkan suatu titik pada diagram alir ke titik lain pada halaman yang sama.
	Penghubung antarhalaman	Digunakan untuk menghubungkan suatu titik pada diagram alir ke titik lain pada halaman yang berbeda. Digunakan apabila diagram lain cukup kompleks sehingga tidak dapat digambar dalam satu halaman.

2 Notasi Algoritma Deskriptif

Notasi algoritma deskriptif dilakukan dengan cara menuliskan intruksi-intruksi yang harus dilaksanakan dalam bentuk uraian kalimat deskriptif dengan menggunakan bahasa yang jelas dan tidak ambigu. Notasi jenis ini disarankan untuk algoritma yang pendek karena apabila untuk algoritma yang panjang notasi deskriptif kurang efektif.

3 Pseudocode

Pseudocode (kode semu atau kode pseudo) adalah suatu bahasa buatan manusia yang sifatnya informal untuk merepresentasikan algoritma. *Pseudocode* dibuat untuk menutupi kekurangan diagram alir dalam merepresentasikan konsep-konsep pemrograman terstruktur. *Pseudocode* memungkinkan representasi langkah-langkah yang lebih detail dan dekat dengan bahasa pemrograman. Karena sifatnya yang informal, tidak ada aturan khusus dalam standar notasi yang dapat digunakan.

Untuk memahami bagaimana untuk menggambarkan suatu algoritma, pada bagian berikut, diberikan contoh dari beberapa proses berpikir yang telah kalian kenal dituliskan dalam diagram alir, notasi algoritma deskriptif dan *pseudocode*.

a. Menghitung Luas Persegi

Diagram Alir	Notasi Algoritma Deskriptif	Pseudocode
<pre> graph TD START([START]) --> READ[/READ sisi/] READ --> PROCESS[Luas = sisi * sisi] PROCESS --> PRINT[/PRINT luas/] PRINT --> END([END]) </pre>	Mulai Baca data sisi Luas adalah sisi kali sisi Tampilkan luas selesai	Start READ sisi Luas sisi * sisi PRINT luas End

b. Membagi bilangan

Diagram Alir	Notasi Algoritma Deskriptif	Pseudocode
<pre> graph TD START([START]) --> READ[/READ pembilang penyebut/] READ --> DEC{penyebut = 0?} DEC -- Yes --> PRINT1[/PRINT "Penyebut tidak boleh nol"/] DEC -- No --> PROCESS[Hasil=pembilang/penyebut] PRINT1 --> END([END]) PROCESS --> PRINT2[/PRINT hasil/] PRINT2 --> END </pre>	Mulai Baca data pembilang dan penyebut Jika penyebut bernilai nol maka tampilkan "Penyebut tidak boleh nol" Jika penyebut bernilai nol maka lakukan pembagian bilangan Hasil adalah pembilang dibagi penyebut Tampilkan hasil selesai	Start READ pembilang, penyebut if penyebut = 0 , "Penyebut tidak boleh nol" else Hasil pembilang/ penyebut PRINT hasil End

c. Mencari Bilangan Terbesar

Diagram Alir	Notasi Algoritma Deskriptif	Pseudocode
<pre> graph TD START([START]) --> READN[/READ N/] READN --> Decision{N > 0?} Decision -- Yes --> PRINTN[/PRINT N/] PRINTN --> Nminus1[N = N - 1] Nminus1 --> Decision Decision -- No --> END([END]) </pre>	Mulai Baca data N Selama nilai $N > 0$, ulangi Tampilkan nilai N Kurangi nilai N dengan 1 Selesai	Start READ N While $N > 0$ Print N N N-1 End